UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/598,558	06/21/2000	Edwin F. Barry	800.0037	6686

27997 7590 03/29/2004

PRIEST & GOLDSTEIN PLLC
5015 SOUTHPARK DRIVE
SUITE 230
DURHAM, NC 27713-7736

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	15

DATE MAILED: 03/29/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/598,558	Applicant(s) BARRY ET AL.
	Examiner David J. Huisman	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-5 and 7-13* is/are rejected.

7)☒ Claim(s) *6* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 August 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 and 3-13 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: #14. RCE as received on 2/3/2004.

Claim Objections

3. Claim 1 is objected to because of the following informalities: The examiner is not clear as to what the phrase, "the SP by one PE array processor environment" in lines 3-4 of claim 1. This should be reworded in a more clear fashion. Appropriate correction is required.

4. Claim 6 is objected to because of the following informalities: The phrase "...to determine whether which register files the SP's register files or the PE's register files are to be accessed..." is not grammatically correct. This phrase should be reworded. A possible suggestion would be to replace "determine whether which register files" with --determine if/whether either--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claim 1 recites the limitations "the SP register file" in line 6, and "the PE register file" in line 7. There is insufficient antecedent basis for this limitation in the claim.

Withdrawn rejections

7. Through amendment, applicant has overcome the rejections set forth in the Office Action mailed on November 3, 2003, for claims 1-13. However, upon further consideration, a new ground(s) of rejection is made below.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1, 7-8, and 11 are rejected under 35 U.S.C. 102(b) as being anticipated by Lee et al., U.S. Patent No. 4,763,242 (as disclosed by applicant and provided in the previous Office Action and herein referred to as Lee).

10. Referring to claim 1, Lee has taught a merged sequence processor (SP) and processor element (PE) processor environment comprising:

a) a first set of registers stored in the SP register file. See Fig.1, component 119.

b) a second set of registers stored in the PE register file. See Fig.3, component 339.

c) a sequence processor/processing element (SP/PE) selection bit in an instruction. See column 4, lines 48-54. Note that the processor field indicates whether the processing element 103 will be used or whether a sequence processor (assist processor) will be used.

d) a context select bit (CSB) in a processor state register, a specific instruction out of the plurality of instruction setting the CSB, the CSB in conjunction with the SP/PE selection bit

selecting a context of a first software task utilizing the first array configuration or a context of a second software task utilizing a second array configuration, the first array configuration including at least one register from the second set of registers to execute sequential instructions, the second array configuration including at least one register from the first set of registers to execute sequential instructions. See column 4, lines 66-68. Note that if the processor field of the instruction indicates that an assist processor will be used, the assist field (CSB field) then selects a context. Furthermore, the instruction itself is inherently stored within an instruction register (processor state register). The instruction register (IR) is an integral component within a processor that holds the instruction that is to be decoded and later executed. When the instruction is in the IR, the context select bit is also in the IR, since the bit is part of the instruction. Consequently, the context select bit is in a processor state register. More specifically, one particular field of the IR will correspond to the CSB field, and each specific instruction, when loaded into the IR, will set that field. In addition, either the main processor or one of multiple coprocessors (COPs) 109 will perform the execution based on the CSB and SP/PE fields. These fields determine which array configuration will be used. For instance, each COP contains its own register file. See Fig.3, component 339. If a COP instruction is encountered, then the CSB and SP/PE fields of the instruction will select a first array configuration, wherein at least one register from the second set of registers is used to execute instructions, i.e., the COP register file will be used while the COP is executing the instruction. On the other hand, if a main processor instruction is encountered, then the CSB and SP/PE fields of that instruction will select a second array configuration, wherein at least one register from the

first set of registers is used to execute instructions, i.e., register file 119 will be used while the main processor is executing the instruction.

11. Referring to claim 7, Lee has taught apparatus as described in claim 1. Lee has further taught that the first or second register files may comprise reconfigurable compute register files (CRF), address register files (ARF), miscellaneous register files (MRF) or a combination of CRF, ARF and MRF files. Note that the first register file 119 contains registers (CRF) used for basic operations (column 4, lines 48-54) and that register file 119 also includes address registers (ARF) which store the addresses of assist instructions and coprocessor configuration registers (MRF). See column 7, lines 1-23.

12. Referring to claim 8, Lee has taught apparatus for providing efficient context switching between tasks in an array of multiple processors including a sequence processor (SP) and multiple processing elements (PE), said apparatus comprising:

a) a first set of registers stored in a first register file for the SP. See Fig.1, component 119.

b) additional sets of registers stored in a plurality of additional register files, with one of the additional sets of registers for each of the PEs. See Fig.3, component 339. Also, note that a plurality of COPs could exist (as shown in Fig.3), each having its own register file.

c) a sequence processor/processing element (SP/PE) selection bit in an instruction. See column 4, lines 48-54. Note that the processor field indicates whether the sequence processor 103 will be used or whether a processing element (assist processor) will be used.

d) a software controllable context select bit (CSB) in a processor state register which in a logical combination with the SP/PE selection bit reconfigures the array by selecting a first context in which the array is configured in a first configuration which provides sequential instructions to

utilize one of the plurality of additional register files or a second context in which the array is configured in a second configuration utilizing the first set of registers for sequential instructions. See column 4, lines 66-68. Note that if the processor field of the instruction indicates that an assist processor will be used, the assist field (CSB field) then selects a context. Furthermore, the instruction itself is inherently stored within an instruction register (processor state register). The instruction register (IR) is an integral component within a processor that holds the instruction that is to be decoded and later executed. When the instruction is in the IR, the context select bit is also in the IR, since the bit is part of the instruction. Consequently, the context select bit is in a processor state register. More specifically, one particular field of the IR will correspond to the CSB field, and each specific instruction, when loaded into the IR, will set that field. In addition, either the main processor or one of multiple coprocessors (COPs) 109 will perform the execution based on the CSB and SP/PE fields. These fields determine which array configuration will be used. For instance, each COP contains its own register file. See Fig.3, component 339. If a COP instruction is encountered, then the CSB and SP/PE fields of the instruction will select a first array configuration, wherein at least one register from the additional sets of registers is used to execute instructions, i.e., the COP register file will be used while the COP is executing the instruction. On the other hand, if a main processor instruction is encountered, then the CSB and SP/PE fields of that instruction will select a second array configuration, wherein at least one register from the first set of registers is used to execute instructions, i.e., register file 119 will be used while the main processor is executing the instruction.

13. Referring to claim 11, Lee has taught a method for providing efficient context switching in an array processor having a sequence processor (SP) and a plurality of processing elements

(PEs), the sequence processor having an SP register file (see Fig.1, component 119), each PE having a PE register file (see Fig.3, component 339), the method comprising:

a) providing a sequence processor/processing element (SP/PE) selection bit in a first instruction. See column 4, lines 48-54. Note that the processor field indicates whether the sequence processor 103 will be used or whether a processing element (assist processor) will be used.

b) setting a context select bit (CSB) in a processor state register with a second instruction. See column 4, lines 66-68. Note that if the processor field of the instruction indicates that an assist processor will be used, the assist field (CSB field) then selects a context. Furthermore, the instruction itself is inherently stored within an instruction register (processor state register). The instruction register (IR) is an integral component within a processor that holds the instruction that is to be decoded and later executed. When the instruction is in the IR, the context select bit is also in the IR, since the bit is part of the instruction. Consequently, the context select bit is in a processor state register. More specifically, one particular field of the IR will correspond to the CSB field, and each specific instruction, when loaded into the IR, will set that field.

c) utilizing the SP/PE selection bit in the first instruction in conjunction with the context select bit stored in the processor state register to determine a context for operation. These bits determine which context/register file will be used. For instance, the COP contains its own register file. See Fig.3, component 339. If a COP instruction is encountered then the COP register file will be used while the COP is executing the instruction. On the other hand, if the main processor is executing the instruction, then its register file 119 will be accessed. Therefore, a register context is selected by the processor field (SP/PE bit) and assist field (CSB) of the instruction.

d) configuring the array processing to have either a first configuration or a second configuration depending upon the context, the first configuration including at least one register file of the plurality of PE register files for a sequential instruction, the second configuration including the SP register file for a sequential instruction. Either the main processor or one of multiple coprocessors (COPs) 109 will perform the execution based on the CSB and SP/PE fields. These fields determine which array configuration will be used. For instance, each COP contains its own register file. See Fig.3, component 339. If a COP instruction is encountered, then the CSB and SP/PE fields of the instruction will select a first array configuration, wherein at least one register of the plurality of PE register files is used to execute instructions, i.e., the COP register file will be used while the COP is executing the instruction. On the other hand, if a main processor instruction is encountered, then the CSB and SP/PE fields of that instruction will select a second array configuration, wherein at least one register from the SP register file is used to execute instructions, i.e., register file 119 will be used while the main processor is executing the instruction.

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 3-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, as applied above, in view of Dowling, U.S. Patent No. 6,128,728 (as applied in the previous Office Action).

16. Referring to claim 3, Lee has taught apparatus as described in claim 1. Lee has not explicitly taught means for allowing the first set of registers to be saved and restored from memory in the background while a task is using the second set of registers in the foreground; and for allowing the second set of registers to be saved and restored from memory in the background while a task is using the first set of registers in the foreground. However, Dowling has taught such a concept. See column 4, lines 32-46, and column 5, line 14, to column 6, line 6. With such a scheme, the inactive register set can be loaded and stored in the background while the active register set can be manipulated by the processor in the foreground. This would maximize the efficiency by making use of otherwise unused external memory cycles, as disclosed in the abstract. That is, when the processor is not accessing memory, the inactive register set can be loaded or stored in the background during those unused memory cycles. Therefore, in order to increase efficiency, it would have been obvious to one of ordinary skill in the art at the time of the invention to save and restore a second register set from memory in the background while a task is using a first register set in the foreground.

17. Referring to claim 4, Lee in view of Dowling has taught apparatus as described in claim 3. Dowling has further taught that said means for allowing comprises a pair of background address registers to provide store and load addresses. See Fig.7, components 780 and 790. Also, for a brief description, see column 19, lines 2-23 (it has been noted that the reference numbers of

the specification do not match all of the numbers in Fig.7, but this should be recognized by one of ordinary skill in the art).

18. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, as applied above, in view of Dowling, U.S. Patent No. 6,170,051 (as applied in the previous Office Action and herein referred to as Dowling2).

19. Referring to claim 5, Lee has taught apparatus as described in claim 1. Lee has further taught a plurality of execution units (see Fig.1, components 145, 117, and 137, for instance) but has not explicitly taught a multiplexer connected to select which registers the execution units read data from and write data to, the multiplexer controlled by a logical combination of the SP/PE selection bit and the CSB. However, Dowling has taught such a concept. See Fig.3, component 320. A person of ordinary skill in the art would have recognized that within Lee's system, only a single register file can be accessed for a particular instruction, and this register file is determined by the SP/PE (processor field) and CSB (assist field). Therefore, since a multiplexer provides a means for selecting between multiple sources (in this case, register files), then it would have been obvious to one of ordinary skill in the art at the time of the invention to use a multiplexer to control access to a particular register file based on the SP/PE and CSB bits.

20. Claims 8-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dahl et al., U.S. Patent No. 5,710,938 (as applied in the previous Office Action and herein referred to as Dahl), in view of Lee, as applied above.

21. Referring to claim 8, Dahl has taught an array multiple processor environment in which a sequence processor (SP) and multiple processing elements (PE) are employed (see Fig.1, components 21 and 12). Dahl has not taught that the array environment includes apparatus for providing efficient context switching between tasks. However Lee has taught such an apparatus, comprising:

a) a first set of registers stored in a first register file for the SP. See Fig.1, component 119.

b) additional sets of registers stored in a plurality of additional register files, with one of the additional sets of registers for each of the PEs. See Fig.3, component 339. Also, note that a plurality of COPs could exist (as shown in Fig.3), each having its own register file.

c) a sequence processor/processing element (SP/PE) selection bit in an instruction. See column 4, lines 48-54. Note that the processor field indicates whether the sequence processor 103 will be used or whether a processing element (assist processor) will be used.

d) a software controllable context select bit (CSB) in a processor state register which in a logical combination with the SP/PE selection bit reconfigures the array by selecting a first context in which the array is configured in a first configuration which provides sequential instructions to utilize one of the plurality of additional register files or a second context in which the array is configured in a second configuration utilizing the first set of registers for sequential instructions. See column 4, lines 66-68. Note that if the processor field of the instruction indicates that an assist processor will be used, the assist field (CSB field) then selects a context. Furthermore, the instruction itself is inherently stored within an instruction register (processor state register). The instruction register (IR) is an integral component within a processor that holds the instruction that is to be decoded and later executed. When the instruction is in the IR, the context select bit

is also in the IR, since the bit is part of the instruction. Consequently, the context select bit is in a processor state register. More specifically, one particular field of the IR will correspond to the CSB field, and each specific instruction, when loaded into the IR, will set that field. In addition, either the main processor or one of multiple coprocessors (COPs) 109 will perform the execution based on the CSB and SP/PE fields. These fields determine which array configuration will be used. For instance, each COP contains its own register file. See Fig.3, component 339. If a COP instruction is encountered, then the CSB and SP/PE fields of the instruction will select a first array configuration, wherein at least one register from the additional sets of registers is used to execute instructions, i.e., the COP register file will be used while the COP is executing the instruction. On the other hand, if a main processor instruction is encountered, then the CSB and SP/PE fields of that instruction will select a second array configuration, wherein at least one register from the first set of registers is used to execute instructions, i.e., register file 119 will be used while the main processor is executing the instruction.

A person of ordinary skill in the art would have recognized that by implementing the concept of context switching within Dahl's system, Dahl could assign different tasks to different processing elements, where each task deals with its own register file contents (note that each task has its own context so in essence, as fetched instructions signify a new task, the context is switched). This allows for an instruction set extension, as disclosed in the abstract, which may allow for increased functionality. Context switching also allows for the executing a second task when a first task is idle for instance. Therefore, in order to increase efficiency, it would have been obvious to one of ordinary skill in the art at the time of the invention to provide an apparatus for context switching (as described in Lee) in the array-processing environment of Dahl.

22. Referring to claim 9, Dahl in view of Lee has taught apparatus as described in claim 8. Dahl has not explicitly taught that the array is a 1x2 array and said first configuration is a 1x2 and said second configuration is a 1x1. However, Dahl has taught the general concept of reconfiguring an array into subarrays. See column 1, lines 42-51. This is desirable because each subarray could load and run separate applications without interaction with other subarrays. This, in turn, would greatly increase the usefulness of the overall array. Although Dahl has not explicitly disclosed the dimensions of the array, a change in size/range is not generally given patentable weight or it would have been considered an obvious improvement. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have a 1x2 array where the first configuration in a 1x2 and the second configuration is a 1x1.

23. Referring to claim 10, Dahl in view of Lee has taught apparatus as described in claim 8. Dahl has not explicitly taught that the array is a 1x5 array and said first configuration is a 1x5 and said second configuration is a 2x2. However, Dahl has taught the general concept of reconfiguring an array into subarrays. See column 1, lines 42-51. This is desirable because each subarray could load and run separate applications without interaction with other subarrays. This, in turn, would greatly increase the usefulness of the overall array. Although Dahl has not explicitly disclosed the dimensions of the array, a change in size/range is not generally given patentable weight or it would have been considered an obvious improvement. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have a 1x5 array where the first configuration in a 1x5 and the second configuration is a 2x2.

24. Claims 12-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, as applied above, and further in view of Mirsky et al., U.S. Patent No. 5,915,123 (as applied in the previous Office Action and herein referred to as Mirsky).

25. Referring to claim 12, Lee has taught a method as described in claim 11.

a) Lee has not taught identifying each PE with both a virtual identifier and a physical identifier. However, Mirsky has taught such a concept. See column 7, lines 29-53 and the abstract. A physical identification allows for the selection of individual processing elements while the virtual identification allows for the selection of one or more processing elements based on a programmable identifier, allowing the software to design its own address space. These different options result in increased flexibility for the programmer. See column 8, lines 6-18. Therefore, in order to increase programmer flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to identifying each PE with both a virtual identifier and a physical identifier, as taught by Mirsky.

b) Lee has not taught identifying each PE utilizing its physical identifier in a first context and identifying each PE utilizing its virtual identifier in a second context. However, Mirsky has shown that data transmitted to the processing elements is used in selecting whether the physical or virtual identification is used. See column 13, lines 27-30, and column 7, lines 47-53. From this it can be seen that each PE can be identified using its physical identifier in a first context (if the transmitted data associated with the first context specifies physical identification) and each PE can be identified using its virtual identifier in a second context (if the transmitted data associated with the second context specifies virtual identification). With this scheme, each context can identify PEs in a way that is most convenient for that context. For instance, virtual

identification may be best for the second context if that context requires the selection of multiple PEs, while if the first context only requires one PE, then physical identification would suffice. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to identify each PE utilizing its physical identifier in a first context and identify each PE utilizing its virtual identifier in a second context.

26. Referring to claim 13, Lee in view of Mirsky has taught a method as described in claim 12. Lee has further taught that the first context is when the CSB bit is inactive and the second context is when the CSB bit is active. Recall from the rejection of claim 11 that the assist field of an instruction is interpreted as performing the same function as the CSB bit. When the assist field is inactive, i.e., if a non-basic instruction is encountered and the assist field must be used to map the instruction to a specific coprocessor (COP), then the selected register file will be the register file associated with the selected COP. This is because each COP has its own register file (context), as shown in Fig.3. On the other hand, if the assist field is active, i.e., if a basic instruction is encountered, the main processor (SP) will perform the execution and therefore, the selected register file will be the main processor's register file. See column 2, lines 57-58. Therefore, it can be seen that the CSB allows for a first context, within the COP, when it is inactive, and a second context, within the main processor, when it is active.

Allowable Subject Matter

27. Claim 6 is objected to as being dependent upon a rejected base claim, but would be allowable if all of its objections are overcome and if it were rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

28. Applicant's arguments filed on February 3, 2004, have been fully considered but they are not persuasive.

29. In the remarks, Applicant argues the novelty/rejection of claims 1, 8, and 11 on page 13 of the remarks, in substance that:

> "Lee does not teach and does not render obvious such techniques for sharing register files. In fact, Lee teaches away from sharing of register files by always using the register file within the main processor or an assist, whichever executes the instruction."

30. These arguments are not found persuasive for the following reasons:

a) It is noted that the features upon which applicant argues (i.e., the sharing of register files) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). The examiner asserts that the claims merely claim selecting one of multiple array configurations, where different configurations use different register files. However, nothing is mentioned about sharing or, for instance, when a PE instruction is executing, the SP register file is used, and vice versa. Therefore, the examiner asserts that Lee still anticipates applicant's claims.

31. In the remarks, Applicant argues the novelty/rejection of claims 1, 8, and 11 on pages 13-14 of the remarks, in substance that:

> "The Official Action misinterprets the role of an instruction register (IR) as described by Hamacher."

32. These arguments are not found persuasive for the following reasons:

a) In general, the applicant has argued that an IR holds an instruction while a processor state register holds data that persists between instructions. The examiner asserts that such a register was not misinterpreted. Instead, the applicant's processor state register is being read in the broadest possible way, which is merely a register that specifies the state of a processor. Clearly, when an instruction is in the IR, the IR's bits are set to those used to encode the instruction, and therefore represent the state of the system when that instruction is to be executed.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH

David J. Huisman
March 15, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100